

文章编号: 2095-2163(2021)04-0114-04

中图分类号: TP311

文献标志码: A

基于 ARKit 的博物馆室内定位系统研究

夏延哲

(广东工业大学 计算机学院, 广州 510006)

摘要: 本文针对博物馆中,不同艺术品摆放比较分散,导致参观者无法及时找到想要欣赏的艺术品的问题,提出制作一种 ARKit 的博物馆室内导航系统,为游览者提供精确的室内导航,提供引导和指引服务。测试结果表明,在系统中给不同的艺术品进行位置信息创建后,可以实现较为准确的路径规划和 AR 导航功能。

关键词: 室内定位系统; WIFI 定位技术; 智能手机

Indoor Navigation in Museums Based on ARKit Technology

XIA Yanzhe

(School of Computer, Guangdong University of Technology, Guangzhou 510006, China)

[Abstract] In this paper, in view of the museum, different artworks are relatively scattered, leading to the problem that visitors cannot find the works of art they want to appreciate in time, put forward the production of an ARKit museum indoor navigation system, for visitors to provide accurate indoor navigation, provide guidance and guidance services. The test results show that the system can achieve more accurate path planning and AR navigation functions after creating location information for different artworks.

[Key words] Indoor navigation; WIFI positioning technology; Smart phone

0 引言

随着中国经济的快速增长,人们不仅仅只满足于物质上的富足,而对精神充实的追求也越来越高。从 2006 年起,中国文物机构和博物馆观览人数持续上升。博物馆已经不仅是收藏、保护、研究、展示文化遗产的机构,还将成为面向未来的文化服务和教育机构。而当游客难以找到参观的艺术品时,就会造成无序的人流流动,导致游客的参观过程非常混乱。

本文提出通过 ARKit 技术,解决传统的博物馆参观中寻找目标艺术品的途径。针对现有室内导航过程中存在的设备运动跟踪不准确、摄像头图像采集、图像视觉处理、场景渲染等技术问题,提出了采用 ARKit 结合 Unity 3D 平台进行室内场景导航的方法;通过整合设备摄像头图像信息与设备运动传感器(包括 LiDAR)信息,在 AR 跟踪稳定性、渲染真实度、人机交互自然度上达到较好的效果。创建前端页面与后端程序的交互,开发室内三维场景导航程序,完成定位场景中坐标点、路径规划、地图信息保存及自动寻路等主要功能。结果证明,该方法可以实现室内自动导航,减少了游客寻找目标展品的时间,提升改善了博物馆的参观游览情况,符合现代化

博物馆建设的需求。

1 ARKit 导航技术

2017 年,苹果公司正式推出了增强现实的开发框架 ARKit。从本质上,AR 是将 2D/3D 元素(文字、图片、模型、音视频等)放置于设备摄像头所采集的图像中,营造一种虚拟元素真实存在于现实世界中的假象。ARKit 通过移动设备(包括手机、平板电脑等)单目摄像头采集的图像信息(包括 LiDAR 采集的信息),实现了平面检测识别、场景几何、环境光估计、环境光反射、图像识别、3D 物体识别等功能。但一个完备的室内导航系统还应该包含前端页面的操作,用户操作流程等。因此,本文提出了将 ARKit 与三维集成平台结合,设计出相应的前端页面,利用 ARKit 的强大功能,通过用户选定目标点,采用合适的算法进行路径规划,最终实现导航的功能。创建一个现实和虚拟空间之间的对应关系,ARKit 使用的技术称为视觉惯性测距。这个过程结合了从 iOS 设备的运动传感硬件与设备的摄像头获取到的图像^[1]。ARKit 识别具有显著特征的场景图像,跟踪视频帧之间的特征差异的位置,将该信息与运动传感数据比较,从而得到一个高精度的设备位置和运动模型,其中包含世界跟

作者简介: 夏延哲(1997-),男,硕士研究生,主要研究方向:室内定位方法和增强现实导航。

收稿日期: 2020-12-29

踪还分析和理解场景的内容。使用命中测试方法找到对应于相机图像中某个点的真实世界的表面。

ARKit 官方提供了 Xcode 开发包,这也提高了利用其它工具进行开发的学习成本。在这种情况下,Unity 构建了一个 AR 开发平台-ARFoundation,该平台架构于 ARKit 和 ARCore 之上,其目的是利用 Unity 的跨平台能力,构建一种与平台无关的 AR 开发环境。即,ARFoundation 对 ARKit 与 ARCore 进行了再次封装,并按照用户的发布平台,自动选择合适的底层 SDK 版本。ARFoundation 提供了一个独立于平台的脚本 API 和 MonoBehaviour。因此,开发者可以通过 ARFoundation 使用 ARCore 和 ARKit 共有的核心功能构建,同时适用于 iOS 和 Android 两个平台的 AR 应用程序。这可以让开发者只需开发一次应用,就可以部署到两个平台的设备上,不必做任何改动^[2]。

2 定位系统的设计与实现

程序定位导航原理的大致流程为:首先进行地图信息的采集;然后对目标特征点信息进行识别和保存;选用合适的路径规划算法,生成导航路径。

2.1 地图信息的采集模块

对真实环境中的扫描和测量,并且保存到后台系统中是该模块重点实现的问题。本程序对于三维环境动态的实时理解,主要依赖于单目 SLAM 技术来实现。在游客移动过程中,一方面要把摄像头所拍到的图像连接成地图,另一方面将地图的信息存储到后方的数据库中。同时,针对如何识别摄像头中的平面问题,采取通过光学系统检测到的特征计算生成。通常称这些点为“点云(Point cloud)”。形成稀疏点云之后,在少量的内存和 CPU 的支持,辅以惯性系统的配合下可以定义成平面,即使在只有少量追踪点的情况下,仍然可以有效地运作^[3]。具体操作流程如图 1 所示。

2.2 艺术品位置信息的采集

此模块的功能主要是对艺术品具体位置进行定位,将艺术品的位置信息传输到服务器。此时需将摄像头中艺术品的位置创建三维物体对象,并且生成该对象在导航程序中全局坐标系数据。只要选定设置好不同艺术品的位置,就可以实现相互之间的路径生成和线路导航。三维物体在场景中的生成如图 2 所示。

产生位置信息具体封装代码如下:

```
void Start()
```

```
{
FindObjectOfType<UIController>( ).Object Information.
Add( transform.position );
FindObjectOfType<UIController>( ). AddPointName
( );
}
```

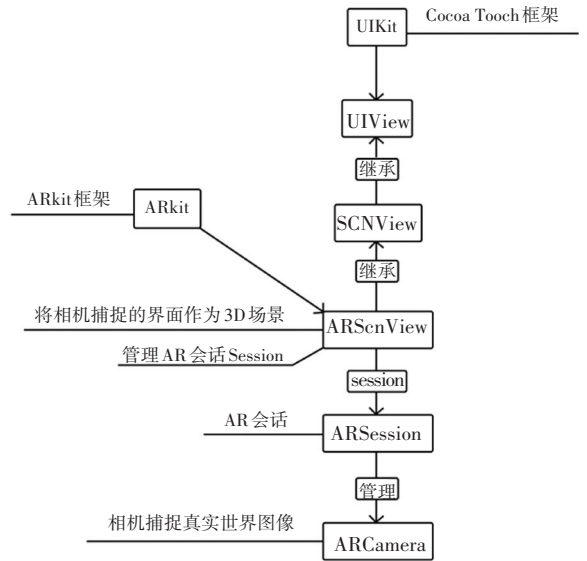


图 1 空间信息识别流程

Fig. 1 Spatial information identification process

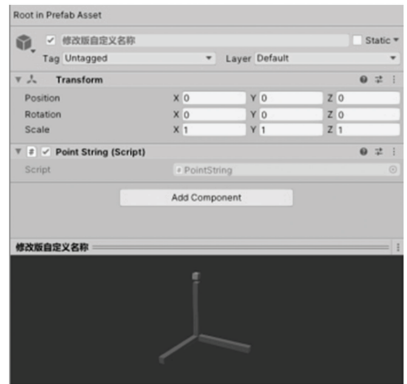


图 2 位置信息采集

Fig. 2 Location information collection

2.3 导航线段的生成

导航路线规划系统是通过两个目标点在地图中的位置信息,来进行规划和确定的。如图 2 所示,首先判断两点是否在同一联通分量上。通过一次完整的 dfs,可以得到一个联通分量,只要两点都被同一次 dfs 访问过,则就存在路径。产生路径的核心算法负责从未被访问到的节点列表 (from 表) 中提取节点的序列号,从终节点的序列号开始一直调用 from[终节点] 往前推,直到推出某个节点 i 的 from[i] = -1, 表示节点 i 没有前节点,即节点 i 就

是起始源节点。虽然推出来的序列号顺序是反向的路径,但可以把这个反向的节点序列压入一个栈中,然后一个一个出栈放入一个向量 *vec* 中保存,这个向量中保存的序列就是正确的顺序。*showPath* 主要调用 *path* 函数来准备路径数据,并传入一个向量 *vector* 给 *path* 函数,用来存储路径信息。之后再控制输出格式把节点一个一个输出来,就形成了一条路径,如图3所示。

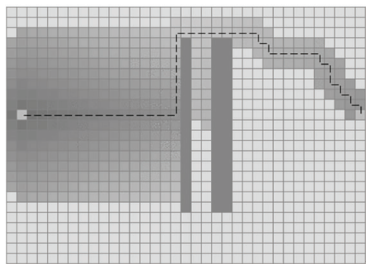


图3 路径规划示意图

Fig. 3 Schematic diagram of path planning

路径规划中的关键代码如下:

```
void dfs(int v) { //传入的 v 为未被访问的节点
    visited[v] = true; //设置传入节点 visited 状态为 true
    tyPename Graph::adjlterator adj(G, v); //提取出图中 v 节点所有邻边的迭代器
    for(int i = adj.begin(); ! adj.end(); i = adj.next()) {
        If(! visited[i]) {
            form[i] = v; //此时访问状态从 v 节点跳到了与其相邻节点之一的 i 节点
            dfs(i); //继续对 i 节点进行深度搜索
        }
    }
}
```

3 导航系统程序开发与发布

3.1 开发平台与开发环境

程序开发语言为.NET C#,其版本为 Visual Studio 2019,本程序支持在 IOS 平台上进行,兼容 IOS12 及以上的操作系统。

3.2 导航系统的开发流程

本程序的开发是以 Unity3D 为主要开发平台,对于程序中需要使用到的脚本文件,则利用 Visual Studio 2019 进行脚本的编辑与测试,在完成脚本的编写与 ARKit 的控制器成功引入之后,将程序发布到 IOS 设备中,并可以正常使用。其具体实现步骤如下:

(1) ARKit 中控制器导入。将 ARKit 提供的一些关键的控制器的引入到程序的工程中。ARWorldMap 控制器用来添加环境扫描,用于扫描摄像头中的空间信

息,并且将当前环境数据保存在本地。AR Session Origin 为创建的地图设置本地坐标信息。

(2) 场景内物体生成。通过 Anchor Prefab 脚本,使得用户点击摄像头中某个已经扫描出来的锚点时,会在本地生成一个 3D 物体,并且记录其当前的位置信息。该 3D 物体的形状和外形可以自定义。

(3) 导航路径规划。在场景中设置 Path Finder 脚本,实现各个目标点之间路径的规划。场景中的任意物体,需要去添加每个节点的坐标,并且在 Paths 中设置每条线路的规划。如:A->B; A->C; B->C; C->D; D->A。

(4) 导航线段的生成与渲染。该部分需要搭配 Unity 中的 LineRenderer 组件来实现。在线段的材质和贴图方面,需要调整几项参数;动态修改 Tiling 参数的信息。当路程较长时,则 X 轴的信息为总路程: MyPoint + for (int i = 0; i < Each Coordinate Information)。将每个坐标的间隔距离相加,就可以得到最终的总路程信息。

(5) 程序 UI 界面的实现。在 UI 设计中,考虑到不同设备的机型,通过微调 UI 面板的宽度与高度信息,来达到系统的自动适配。当设计分辨率与设备分辨率不同时,需要将 UI 界面经过一定缩放投射到屏幕上。

(6) 程序的调试以及 bug 的修改。

(7) 撰写程序的使用说明以及开发文档。

3.3 程序 UI 及窗体设计、发布设置

一个优秀的导航程序,离不开友好、人性化的使用界面。程序的界面和窗体设计是用户是否能快速上手使用的重要因素。本文导航程序使用 FairyGUI 来设计前端操作页面,通过前端清晰的操作按钮与后台控制器的绑定,实现了程序操作的简化,同时提高了易用性。程序的主要功能以及定位导航过程示意图如图4所示。

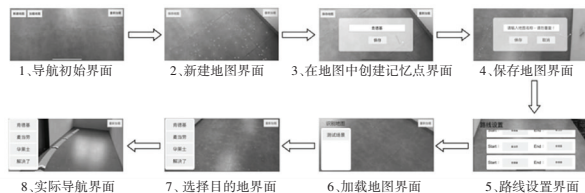


图4 定位导航过程示意图

Fig. 4 Schematic diagram of positioning and navigation process

4 结束语

本文根据目前博物馆参览的实际情况,在对相 (下转第 123 页)